Data Cleaning

Sally Chen

Contents

```r
library(dplyr)
library(fpp)
library(fpp2)
library(seasonal)
library(ggplot2)
library(imputeTS)  # imputation for ts
```

```r
dailyweather = read.csv("DailyWeather.csv")
colnames(dailyweather)
```

```
## [1] "Year"     "Month"    "Day"      "sunshine" "wind"     "rainfall" "max"
## [8] "min"
```

Data exploration

```r
summary(dailyweather)
```

```
##       Year          Month             Day           sunshine           wind
##  Min.   :1970   Min.   : 1.000   Min.   : 1.00   Min.   : 0.000   Min.   :  5.40
##  1st Qu.:1982   1st Qu.: 4.000   1st Qu.: 8.00   1st Qu.: 3.300   1st Qu.: 35.30
##  Median :1995   Median : 7.000   Median :16.00   Median : 6.800   Median : 44.60
##  Mean   :1995   Mean   : 6.529   Mean   :15.72   Mean   : 6.548   Mean   : 47.61
##  3rd Qu.:2007   3rd Qu.:10.000   3rd Qu.:23.00   3rd Qu.: 9.700   3rd Qu.: 59.40
##  Max.   :2020   Max.   :12.000   Max.   :31.00   Max.   :13.900   Max.   :139.00
##                                                  NA's   :10642    NA's   :64
##     rainfall             max               min
##  Min.   :  0.000   Min.   : 5.70   Min.   :-2.500
##  1st Qu.:  0.000   1st Qu.:14.80   1st Qu.: 6.400  
##  Median :  0.000   Median :18.70   Median : 9.200  
##  Mean   :  1.467   Mean   :19.89   Mean   : 9.564  
##  3rd Qu.:  0.800   3rd Qu.:23.70   3rd Qu.:12.400  
##  Max.   :138.800   Max.   :46.80   Max.   :30.500  
##                    NA's   :1       NA's   :18      
```

More than 50% missing value in sunshine variable

Combine date columns

```
df_combinedate = dailyweather %>% mutate(date = as.Date(with(dailyweather, paste(Year,
    Month, Day, sep = "-")), "%Y-%m-%d")) %>% select(-c("Year", "Month", "Day"))
```

Missing values clustering index

```
col_sunshine = df_combinedate$sunshine
x = 0
for (i in 1:length(col_sunshine)) {
    if (is.na(col_sunshine)[i] == FALSE) {
        x = i
        break
    }
}

# sunshine value all equal to NA before 1999-08-18
df_combinedate[x, ]
```

```
##       sunshine wind rainfall  max min       date
## 10641      2.7 87.1        0 15.4 6.1 1999-08-18
```

```
df_remove_naclust = df_combinedate[-(1:x - 1), ]  #df after 1999-08-18
summary(df_remove_naclust)
```

```
##     sunshine          wind           rainfall            max       
##  Min.   : 0.000   Min.   : 13.00   Min.   :  0.000   Min.   : 8.10  
##  1st Qu.: 3.300   1st Qu.: 35.30   1st Qu.:  0.000   1st Qu.:15.30  
##  Median : 6.800   Median : 44.60   Median :  0.000   Median :19.30  
##  Mean   : 6.548   Mean   : 47.32   Mean   :  1.394   Mean   :20.55  
##  3rd Qu.: 9.700   3rd Qu.: 57.60   3rd Qu.:  0.600   3rd Qu.:24.40  
##  Max.   :13.900   Max.   :122.40   Max.   :138.800   Max.   :46.80  
##  NA's   :2        NA's   :25                                        
##       min              date           
##  Min.   :-2.000   Min.   :1999-08-18  
##  1st Qu.: 6.700   1st Qu.:2004-10-15  
##  Median : 9.500   Median :2009-12-13  
##  Mean   : 9.881   Mean   :2009-12-13  
##  3rd Qu.:12.800   3rd Qu.:2015-02-10  
##  Max.   :30.500   Max.   :2020-04-10  
##  NA's   :1                            
```

Remaining missing value index

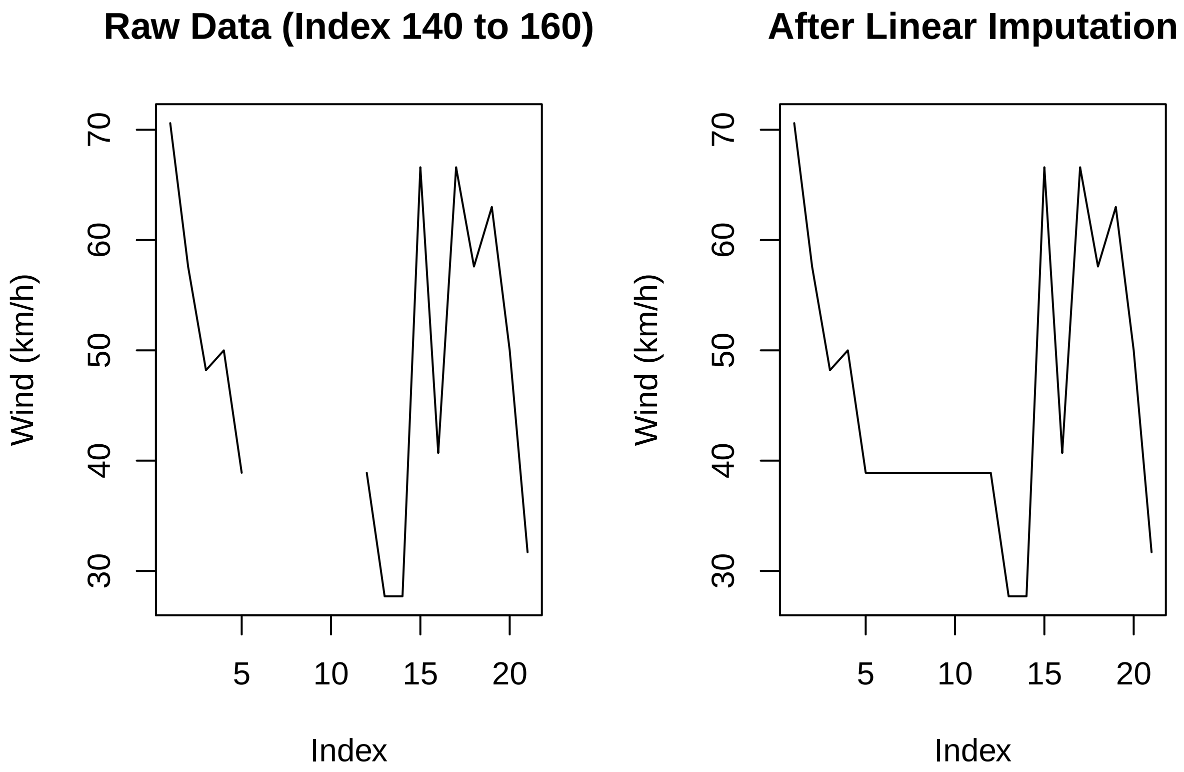
```
## sunshine
col_sunshine1 = df_remove_naclust$sunshine
c = 1
y1 = NA
for (i in 1:length(col_sunshine1)) {
    if (is.na(col_sunshine1)[i] & c <= 2) {
        y1[c] = i
        c = c + 1
    }
}
y1  # 4067 7535


## wind
col_wind = df_remove_naclust$wind
c = 1
y2 = NA

for (i in 1:length(col_wind)) {
    if (is.na(col_wind)[i] & c <= 25) {
        y2[c] = i
        c = c + 1
    }
}
y2  #24  145  146  147  148  149  150 1908 2441 2442 2635 3109 3517 3616 3677 3798 4155
↪  4354 4419 4425 4504 4520 5370 6374 6375

## min
y3 = NA
col_min = df_remove_naclust$min
for (i in 1:length(col_min)) {
    if (is.na(col_min)[i]) {
        y3 = i
        break
    }
}
y3  # 15
```

```
## [1] 4067 7535
##  [1]   24  145  146  147  148  149  150 1908 2441 2442 2635 3109 3517 3616 3677 3798
## [17] 4155 4354 4419 4425 4504 4520 5370 6374 6375
## [1] 15
```

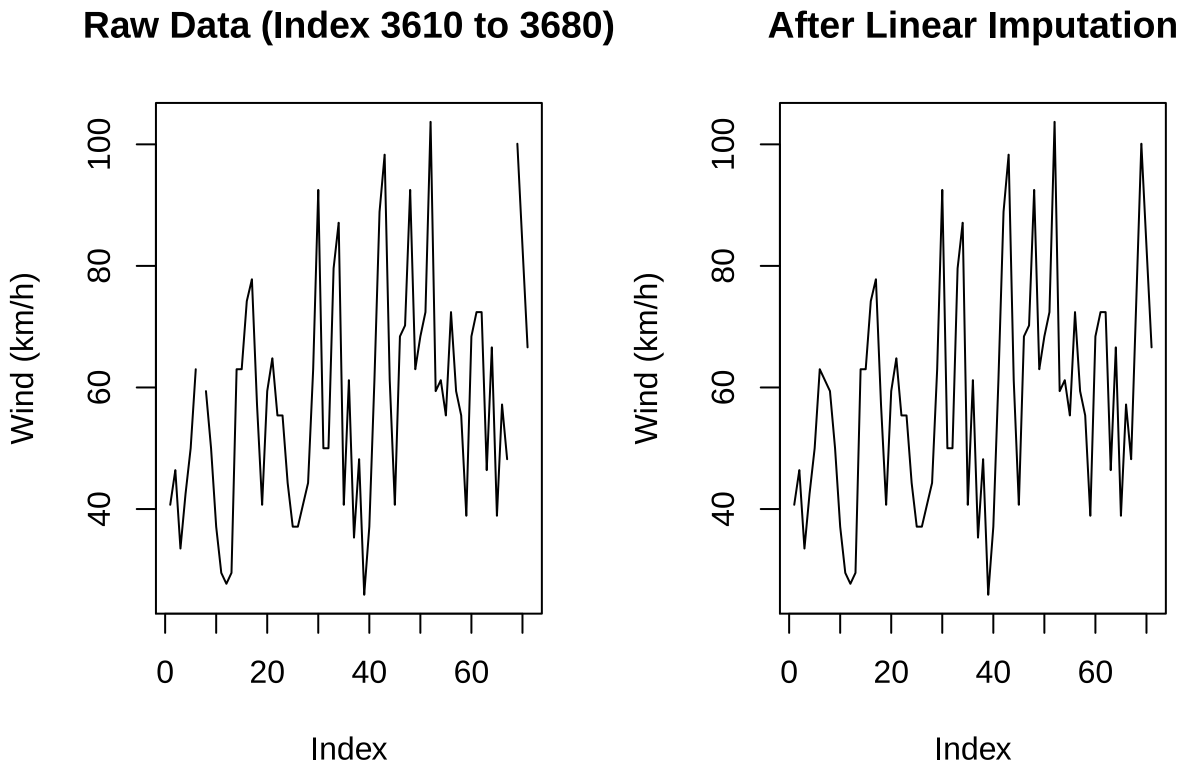
Missing Value Imputation

```
ts_df = ts(df_remove_naclust, frequency = 365.25, start = c(1999, 8))
ts_imputed = na_interpolation(ts_df, option = "linear")  # linear interpolation
```

```
# Check imputation
par(mfrow = c(1, 2))
plot(ts_df[140:160, "wind"], type = "l", ylab = "Wind (km/h)", main = "Raw Data (Index
↪  140 to 160)")
plot(ts_imputed[140:160, "wind"], type = "l", ylab = "Wind (km/h)", main = "After Linear
↪  Imputation")
```



```
par(mfrow = c(1, 2))
plot(ts_df[3610:3680, "wind"], type = "l", ylab = "Wind (km/h)", main = "Raw Data (Index
↪  3610 to 3680)")
plot(ts_imputed[3610:3680, "wind"], type = "l", ylab = "Wind (km/h)", main = "After
↪  Linear Imputation")
```



Train and Test Set Splitting

```r
# split train and test set
split_index = length(ts_imputed[, 6]) * 0.8
lower = ts_imputed[, 6][round(split_index)]
upper = ts_imputed[, 6][round(split_index) + 1]

train = ts_imputed[ts_imputed[, 6] <= lower, ]
test = ts_imputed[ts_imputed[, 6] >= upper, ]
```

```r
# Check test and train set ratio
length(test)/length(ts_imputed)  # 20%
length(train)/length(ts_imputed)  # 80%
length(train) + length(test) == length(ts_imputed)

# Check start and end dates of test data
as.Date(train[, "date"])[6034]
```

```r
train_df = train %>% as.data.frame() %>% mutate(date = as.Date(date))
test_df = test %>% as.data.frame() %>% mutate(date = as.Date(date))

write.csv(train_df, "train.csv", row.names = FALSE)
write.csv(test_df, "test.csv", row.names = FALSE)
```